Ceph存储

环境：centos7.4虚拟机4台，搭建好yum源，关闭防火墙、SELinux，清空iptables规则

所有机器配置好相互免密（包括自己），

做好时间同步（chronyd服务）

做好域名解析（hosts文件）

做好yum源（centos和ceph）

规划： node01 192.168.1.1 ceph1 3 disk 10G

Node02 192.168.1.2 ceph2 3 disk 10G

Node03 192.168.1.3 ceph3 3 disk 10G

Node10 192.168.1.10 client

#部署ceph集群

[root@node01 ~]# yum -y install ceph-deploy.noarch

[root@node01 ~]# mkdir ceph-cluster

[root@node01 ~]# cd ceph-cluster/

[root@node01 ceph-cluster]#

[root@node01 ceph-cluster]# ceph-deploy new node01 node02 node03

[root@node01 ceph-cluster]# ls

ceph.conf ceph-deploy-ceph.log ceph.mon.keyring

[root@node01 ceph-cluster]# for i in node0{1..3}

> do

> ssh $i "yum -y install ceph-mon ceph-osd ceph-mds ceph-radosgw"

> done

。。。。。。

[root@node01 ceph-cluster]# ceph-deploy mon create-initial

[root@node01 ceph-cluster]# ls /etc/ceph/

ceph.client.admin.keyring ceph.conf rbdmap tmpoivlSS

[root@node01 ceph-cluster]# for i in node0{1..3}

> do

> ssh $i "systemctl is-active ceph-mon@$i.service"

> done

active

active

active

[root@node01 ceph-cluster]# ceph -s

cluster ed07c6fa-806d-4b16-806b-47c63fa08853

health HEALTH\_ERR

64 pgs are stuck inactive for more than 300 seconds

64 pgs stuck inactive

no osds

monmap e1: 3 mons at {node01=192.168.1.1:6789/0,node02=192.168.1.2:6789/0,node03=192.168.1.3:6789/0}

election epoch 4, quorum 0,1,2 node01,node02,node03

osdmap e1: 0 osds: 0 up, 0 in

flags sortbitwise

pgmap v2: 64 pgs, 1 pools, 0 bytes data, 0 objects

0 kB used, 0 kB / 0 kB avail

64 creating

[root@node01 ceph-cluster]#

[root@node01 ceph-cluster]# for i in node0{1..3}

> do

> ssh $i "parted /dev/vdb mklabel gpt"

> ssh $i "parted /dev/vdb mkpart primary 1 50%"

> ssh $i "parted /dev/vdb mkpart primary 50% 100%"

> done

[root@node01 ceph-cluster]# ll /dev/vdb?

brw-rw---- 1 root disk 253, 17 12月 19 18:40 /dev/vdb1

brw-rw---- 1 root disk 253, 18 12月 19 18:40 /dev/vdb2

[root@node01 ceph-cluster]# vim /etc/udev/rules.d/80-vdb.rules

[root@node01 ceph-cluster]# cat /etc/udev/rules.d/80-vdb.rules

ENV{DEVNAME}=="/dev/vdb1", OWNER="ceph", GROUP="ceph"

ENV{DEVNAME}=="/dev/vdb2", OWNER="ceph", GROUP="ceph"

[root@node01 ceph-cluster]# for i in node02 node03

> do

> scp /etc/udev/rules.d/80-vdb.rules $i:/etc/udev/rules.d/

> done

#重启系统，验证udev规则是否生效

[root@hostos ~]# for i in node0{1..3}

> do

> virsh destroy $i

> virsh start $i

> done

[root@node01 ~]# for i in node0{1..3}

> do

> echo $i

> ssh $i "ls -l /dev/vdb?"

> done

node01

brw-rw---- 1 ceph ceph 253, 17 12月 20 2019 /dev/vdb1

brw-rw---- 1 ceph ceph 253, 18 12月 20 2019 /dev/vdb2

node02

brw-rw---- 1 ceph ceph 253, 17 12月 20 2019 /dev/vdb1

brw-rw---- 1 ceph ceph 253, 18 12月 20 2019 /dev/vdb2

node03

brw-rw---- 1 ceph ceph 253, 17 12月 20 2019 /dev/vdb1

brw-rw---- 1 ceph ceph 253, 18 12月 20 2019 /dev/vdb2

#清空磁盘

[root@node01 ceph-cluster]# for i in node0{1..3}

> do

> ceph-deploy disk zap $i:vdc $i:vdd

> done

#创建osd存储

[root@node01 ceph-cluster]# for i in node0{1..3}

> do

> ceph-deploy osd create $i:vdc:/dev/vdb1 $i:vdd:/dev/vdb2

> done

[root@node01 ceph-cluster]# ceph -s

cluster ed07c6fa-806d-4b16-806b-47c63fa08853

health HEALTH\_OK

monmap e1: 3 mons at {node01=192.168.1.1:6789/0,node02=192.168.1.2:6789/0,node03=192.168.1.3:6789/0}

election epoch 18, quorum 0,1,2 node01,node02,node03

osdmap e47: 6 osds: 6 up, 6 in

flags sortbitwise

pgmap v89: 64 pgs, 1 pools, 0 bytes data, 0 objects

203 MB used, 61170 MB / 61373 MB avail

64 active+clean

[root@node01 ceph-cluster]#

[root@node01 ceph-cluster]# ceph osd tree

ID WEIGHT TYPE NAME UP/DOWN REWEIGHT PRIMARY-AFFINITY

-1 0.05878 root default

-2 0.01959 host node03

0 0.00980 osd.0 up 1.00000 1.00000

3 0.00980 osd.3 up 1.00000 1.00000

-3 0.01959 host node02

1 0.00980 osd.1 up 1.00000 1.00000

2 0.00980 osd.2 up 1.00000 1.00000

-4 0.01959 host node01

4 0.00980 osd.4 up 1.00000 1.00000

5 0.00980 osd.5 up 1.00000 1.00000

[root@node01 ceph-cluster]#

#查看存储池

[root@node01 ceph-cluster]# ceph osd lspools

0 rbd,

#创建镜像

[root@node01 ceph-cluster]# rbd create image\_one --image-feature layering --size 10G

[root@node01 ceph-cluster]# rbd list

image\_one

[root@node01 ceph-cluster]# rbd info image\_one

rbd image 'image\_one':

size 10240 MB in 2560 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.16fcb238e1f29

format: 2

features: layering

flags:

[root@node01 ceph-cluster]# rbd create image\_two --image-feature layering --size 10G

[root@node01 ceph-cluster]# rbd list

image\_one

image\_two

[root@node01 ceph-cluster]#

#修改镜像大小

[root@node01 ceph-cluster]# rbd resize --size 8G --allow-shrink image\_one

Resizing image: 100% complete...done.

[root@node01 ceph-cluster]# rbd info image\_one

rbd image 'image\_one':

size 8192 MB in 2048 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.16fcb238e1f29

format: 2

features: layering

flags:

[root@node01 ceph-cluster]#

[root@node01 ceph-cluster]# rbd resize --size 15G image\_one

Resizing image: 100% complete...done.

[root@node01 ceph-cluster]# rbd info image\_one

rbd image 'image\_one':

size 15360 MB in 3840 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.16fcb238e1f29

format: 2

features: layering

flags:

[root@node01 ceph-cluster]#

[root@node10 ~]# yum -y install ceph-common.x86\_64

[root@node10 ~]# scp 192.168.1.1:/etc/ceph/\* /etc/ceph/

[root@node10 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 20G 0 disk

└─vda1 253:1 0 20G 0 part /

[root@node10 ~]# rbd map image\_one

/dev/rbd0

[root@node10 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 20G 0 disk

└─vda1 253:1 0 20G 0 part /

rbd0 252:0 0 15G 0 disk

[root@node10 ~]# rbd map image\_two

/dev/rbd1

[root@node10 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 20G 0 disk

└─vda1 253:1 0 20G 0 part /

rbd0 252:0 0 15G 0 disk

rbd1 252:16 0 10G 0 disk

[root@node10 ~]#

[root@node10 ~]# rbd showmapped

id pool image snap device

0 rbd image\_one - /dev/rbd0

1 rbd image\_two - /dev/rbd1

#格式化挂载块设备

[root@node10 ~]# mkfs.xfs /dev/rbd0

[root@node10 ~]# mount /dev/rbd0 /mnt/

[root@node10 ~]# df -hT | grep mnt

/dev/rbd0 xfs 15G 33M 15G 1% /mnt

[root@node10 ~]#

[root@node10 ~]# echo test >> /mnt/test.txt

#镜像快照

[root@node01 ceph-cluster]# rbd snap ls image\_one

[root@node01 ceph-cluster]# rbd snap create image\_one --snap image\_one\_snap1

[root@node01 ceph-cluster]# rbd snap ls image\_one

SNAPID NAME SIZE

6 image\_one\_snap1 15360 MB

#模拟误删数据

[root@node10 ~]# ls /mnt/

test.txt

[root@node10 ~]# rm -rf /mnt/test.txt

#ceph不支持在线恢复快照，必须卸载

[root@node10 ~]# umount /mnt/

#恢复快照

[root@node01 ceph-cluster]# rbd snap rollback image\_one --snap image\_one\_snap1

Rolling back to snapshot: 100% complete...done.

#客户端挂载

[root@node10 ~]# mount /dev/rbd0 /mnt/

[root@node10 ~]# ls /mnt/

test.txt

[root@node10 ~]#

[root@node01 ceph-cluster]# rbd snap ls image\_one

SNAPID NAME SIZE

6 image\_one\_snap1 15360 MB

#给快照加保护，防止克隆时删除快照

[root@node01 ceph-cluster]# rbd snap protect image\_one --snap image\_one\_snap1

#测试删除处于保护状态的快照

[root@node01 ceph-cluster]# rbd snap rm image\_one --snap image\_one\_snap1

rbd: snapshot 'image\_one\_snap1' is protected from removal.

2019-12-20 15:09:58.630979 7fd352e0ad80 -1 librbd::Operations: snapshot is protected

#克隆镜像

[root@node01 ceph-cluster]# rbd clone image\_one --snap image\_one\_snap1 image\_one\_clone --image-feature layering

[root@node01 ceph-cluster]# rbd list

image\_one

image\_one\_clone

image\_two

#克隆出来的镜像基于原始快照，所以原始快照不能删除

[root@node01 ceph-cluster]# rbd info image\_one\_clone

rbd image 'image\_one\_clone':

size 15360 MB in 3840 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.16fef238e1f29

format: 2

features: layering

flags:

parent: rbd/image\_one@image\_one\_snap1

overlap: 15360 MB

#取消快照关联

[root@node01 ceph-cluster]# rbd flatten image\_one\_clone

Image flatten: 100% complete...done.

[root@node01 ceph-cluster]# rbd info image\_one\_clone

rbd image 'image\_one\_clone':

size 15360 MB in 3840 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.16fef238e1f29

format: 2

features: layering

flags:

#删除原始快照

[root@node01 ceph-cluster]# rbd snap unprotect image\_one --snap image\_one\_snap1

[root@node01 ceph-cluster]# rbd snap rm image\_one --snap image\_one\_snap1

[root@node01 ceph-cluster]# rbd snap ls image\_one

#测试克隆镜像

#######原始镜像和克隆镜像在同一个电脑上只能挂载一个######

#利用libvirtd使用ceph块存储

[root@node01 ceph-cluster]# rbd create image\_vm1 --image-feature layering --size 10G

[root@node01 ceph-cluster]# rbd info image\_vm1

rbd image 'image\_vm1':

size 10240 MB in 2560 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.16fdc2ae8944a

format: 2

features: layering

flags:

#查看用户名密码

[root@node01 ceph-cluster]# cat /etc/ceph/ceph.client.admin.keyring

[client.admin]

key = AQBST/tdlTEBJBAALypqAiSDSR/FJe8Qmm7tGA==

#宿主机设置秘钥认证

[root@hostos ~]# vim secret.xml

[root@hostos ~]# cat secret.xml

<secret ephemeral='no' private='no'>

<usage type='ceph'>

<name>

client.admin secret

</name>

</usage>

</secret>

[root@hostos ~]# virsh secret-define secret.xml

生成 secret 4de68538-e822-4c18-a3f7-253184b4b12b

[root@hostos ~]# virsh secret-list

UUID 用量

--------------------------------------------------------------------------------

4de68538-e822-4c18-a3f7-253184b4b12b ceph

client.admin secret

[root@hostos ~]# virsh secret-set-value --secret 4de68538-e822-4c18-a3f7-253184b4b12b --base64 AQBST/tdlTEBJBAALypqAiSDSR/FJe8Qmm7tGA==

secret 值设定

[root@hostos ~]# virsh secret-list

UUID 用量

--------------------------------------------------------------------------------

4de68538-e822-4c18-a3f7-253184b4b12b ceph

client.admin secret

#新建虚拟机测试ceph块存储

[root@hostos ~]# virsh edit node08

23 <disk type='file' device='disk'>

24 <driver name='qemu' type='qcow2'/>

25 <source file='/var/lib/libvirt/images/node08.img'/>

26 <target dev='vda' bus='virtio'/>

27 <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>

28 </disk>

29 <disk type='network' device='disk'>

30 <driver name='qemu' type='raw'/>

31 <auth username='admin'>

32 <secret type='ceph' uuid='4de68538-e822-4c18-a3f7-253184b4b12b'/>

33 </auth>

34 <source protocol='rbd' name='rbd/image\_vm1'>

35 <host name='192.168.1.1' port='6789'/>

36 </source>

37 <target dev='vdb' bus='virtio'/>

38 <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'/>

39 </disk>

[root@hostos ~]# virsh start node08 –console

localhost login: root

Password:

Last login: Fri Dec 20 18:00:42 on ttyS0

[root@localhost ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 20G 0 disk

└─vda1 253:1 0 20G 0 part /

vdb 253:16 0 10G 0 disk

[root@localhost ~]#

#ceph块存储成功加载

#ceph文件系统共享

#新建机器node04，同步yum源，做好chronyd时间同步，做好域名解析，免密登录

[root@node04 ~]# yum -y install ceph-mds

[root@node01 ceph-cluster]# ceph-deploy mds create node04

#同步秘钥文件

[root@node01 ceph-cluster]# ceph-deploy admin node04

#验证服务状态

[root@node04 ~]# systemctl is-active ceph-mds@node04.service

active

[root@node04 ~]# systemctl is-enabled ceph-mds@node04.service

enabled

[root@node04 ~]#

#创建存储池

[root@node01 ceph-cluster]# ceph osd lspools

0 rbd,

[root@node01 ceph-cluster]# ceph osd pool create cephfs\_data 128

pool 'cephfs\_data' created

[root@node01 ceph-cluster]# ceph osd pool create cephfs\_metadata 128

pool 'cephfs\_metadata' created

[root@node01 ceph-cluster]# ceph osd lspools

0 rbd,3 cephfs\_data,4 cephfs\_metadata,

[root@node01 ceph-cluster]#

#定义文件系统##ceph集群只能做一个文件系统

[root@node04 ~]# ceph mds stat

e2:, 1 up:standby

[root@node04 ~]# ceph fs new mycephfs cephfs\_metadata cephfs\_data

new fs with metadata pool 4 and data pool 3

[root@node04 ~]# ceph fs ls

name: mycephfs, metadata pool: cephfs\_metadata, data pools: [cephfs\_data ]

[root@node04 ~]# ceph mds stat

e4: 1/1/1 up {0=node04=up:creating}

[root@node04 ~]#

#客户端挂载

[root@node10 ~]# cat /etc/ceph/ceph.client.admin.keyring

[client.admin]

key = AQBST/tdlTEBJBAALypqAiSDSR/FJe8Qmm7tGA==

[root@node10 ~]# mount -t ceph -o name=admin,secret=AQBST/tdlTEBJBAALypqAiSDSR/FJe8Qmm7tGA== 192.168.1.1:6789:/ /mnt

[root@node10 ~]# df -hT | grep ceph

192.168.1.1:6789:/ ceph 60G 492M 60G 1% /mnt

[root@node10 ~]#

#测试读写

[root@node10 ~]# echo testceph > /mnt/a.txt

[root@node10 ~]# cat /mnt/a.txt

testceph

[root@node10 ~]#

##测试空间占用

[root@node10 ~]# ceph -s | grep used

492 MB used, 60881 MB / 61373 MB avail

[root@node10 ~]# dd if=/dev/zero of=/mnt/b.txt bs=1M count=500

记录了500+0 的读入

记录了500+0 的写出

524288000字节(524 MB)已复制，33.4965 秒，15.7 MB/秒

[root@node10 ~]# du -sh /mnt/

501M /mnt/

[root@node10 ~]#

[root@node10 ~]# ceph -s | grep used

1993 MB used, 59380 MB / 61373 MB avail

[root@node10 ~]#

#由于ceph本身自带的三副本功能，所以存一份500M的数据，实际占用存储空间应该是大于1.5G的

#在node04上部署radosgw

[root@node04 ~]# yum -y install ceph-radosgw

[root@node01 ceph-cluster]# ceph-deploy rgw create node04

[root@node01 ceph-cluster]# ceph-deploy admin node04

[root@node04 ~]# systemctl is-active ceph-radosgw@rgw.node04.service

active

[root@node04 ~]# systemctl is-enabled ceph-radosgw@rgw.node04.service

enabled

[root@node04 ~]#

[root@node04 ~]# netstat -antpu | grep radosgw

tcp 0 0 0.0.0.0:7480 0.0.0.0:\* LISTEN 1957/radosgw

#修改服务端口

[root@node04 ~]# vim /etc/ceph/ceph.conf

[root@node04 ~]# tail -3 /etc/ceph/ceph.conf

[client.rgw.node04]

host = node04

rgw\_frontends = "civetweb port=8000"

[root@node04 ~]# systemctl restart ceph-radosgw@rgw.node04.service

[root@node04 ~]# netstat -antpu | grep rados

tcp 0 0 0.0.0.0:8000 0.0.0.0:\* LISTEN 2331/radosgw

#####客户端软件测试

[root@node10 ~]# curl -I 192.168.1.4:8000

HTTP/1.1 200 OK

[root@node04 ~]# radosgw-admin user create --uid='testuser' --display-name='First User'

{

"user\_id": "testuser",

"display\_name": "First User",

"email": "",

"suspended": 0,

"max\_buckets": 1000,

"auid": 0,

"subusers": [],

"keys": [

{

"user": "testuser",

"access\_key": "Q2XK91IY143WLJA83XCX",

"secret\_key": "mJrPxrjvuexDX7Mp1wajJnrz6uQXtCBy4AOCtxhA"

}

],

"swift\_keys": [],

"caps": [],

"op\_mask": "read, write, delete",

"default\_placement": "",

"placement\_tags": [],

"bucket\_quota": {

"enabled": false,

"max\_size\_kb": -1,

"max\_objects": -1

},

"user\_quota": {

"enabled": false,

"max\_size\_kb": -1,

"max\_objects": -1

},

"temp\_url\_keys": []

}

[root@node10 ~]# yum -y install ./s3cmd-2.0.2-1.el7.noarch.rpm

[root@node10 ~]# s3cmd --configure

Enter new values or accept defaults in brackets with Enter.

Refer to user manual for detailed description of all options.

Access key and Secret key are your identifiers for Amazon S3. Leave them empty for using the env variables.

Access Key: Q2XK91IY143WLJA83XCX

Secret Key: mJrPxrjvuexDX7Mp1wajJnrz6uQXtCBy4AOCtxhA

Default Region [US]:

Use "s3.amazonaws.com" for S3 Endpoint and not modify it to the target Amazon S3.

S3 Endpoint [s3.amazonaws.com]: 192.168.1.4:8000

Use "%(bucket)s.s3.amazonaws.com" to the target Amazon S3. "%(bucket)s" and "%(location)s" vars can be used

if the target S3 system supports dns based buckets.

DNS-style bucket+hostname:port template for accessing a bucket [%(bucket)s.s3.amazonaws.com]: %(bucket)s.192.168.1.4:8000

Encryption password is used to protect your files from reading

by unauthorized persons while in transfer to S3

Encryption password:

Path to GPG program [/usr/bin/gpg]:

When using secure HTTPS protocol all communication with Amazon S3

servers is protected from 3rd party eavesdropping. This method is

slower than plain HTTP, and can only be proxied with Python 2.7 or newer

Use HTTPS protocol [Yes]: No

On some networks all internet access must go through a HTTP proxy.

Try setting it here if you can't connect to S3 directly

HTTP Proxy server name:

New settings:

Access Key: Q2XK91IY143WLJA83XCX

Secret Key: mJrPxrjvuexDX7Mp1wajJnrz6uQXtCBy4AOCtxhA

Default Region: US

S3 Endpoint: 192.168.1.4:8000

DNS-style bucket+hostname:port template for accessing a bucket: %(bucket)s.192.168.1.4:8000

Encryption password:

Path to GPG program: /usr/bin/gpg

Use HTTPS protocol: False

HTTP Proxy server name:

HTTP Proxy server port: 0

Test access with supplied credentials? [Y/n] n

Save settings? [y/N] y

Configuration saved to '/root/.s3cfg'

[root@node10 ~]#

#功能测试

[root@node10 ~]# s3cmd ls

[root@node10 ~]# s3cmd mb s3://my\_bucket

Bucket 's3://my\_bucket/' created

[root@node10 ~]# s3cmd ls

2019-12-26 10:33 s3://my\_bucket

[root@node10 ~]# s3cmd put /var/log/messages s3://my\_bucket/log/

upload: '/var/log/messages' -> 's3://my\_bucket/log/messages' [1 of 1]

150071 of 150071 100% in 5s 25.61 kB/s done

[root@node10 ~]# s3cmd ls

2019-12-26 10:33 s3://my\_bucket

[root@node10 ~]# s3cmd ls s3://my\_bucket

DIR s3://my\_bucket/log/

[root@node10 ~]# s3cmd ls s3://my\_bucket/log/

2019-12-26 10:33 150071 s3://my\_bucket/log/messages

[root@node10 ~]# s3cmd ls s3://my\_bucket/log/messages

2019-12-26 10:33 150071 s3://my\_bucket/log/messages

[root@node10 ~]#

[root@node10 ~]# s3cmd get s3://my\_bucket/log/messages /tmp/

download: 's3://my\_bucket/log/messages' -> '/tmp/messages' [1 of 1]

150071 of 150071 100% in 0s 16.23 MB/s done

[root@node10 ~]# ls /tmp/messages

/tmp/messages

[root@node10 ~]# s3cmd del s3://my\_bucket/log/messages

delete: 's3://my\_bucket/log/messages'

[root@node10 ~]# s3cmd ls s3://my\_bucket/log/

[root@node10 ~]# s3cmd ls s3://my\_bucket/

[root@node10 ~]# s3cmd ls

2019-12-26 10:33 s3://my\_bucket